```
/*
  Vcc - A supply voltage measuring library for Arduino

  Created by Ivo Pullens, Emmission, 2014

  Inspired by:
  http://provideyourown.com/2012/secret-arduino-voltmeter-measure-battery-voltage/

  This library is free software; you can redistribute it and/or
  modify it under the terms of the GNU Lesser General Public
  License as published by the Free Software Foundation; either
  version 2.1 of the License, or (at your option) any later version.

  This library is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
  Lesser General Public License for more details.

  You should have received a copy of the GNU Lesser General Public
  License along with this library; if not, write to the Free Software
  Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
*/

#ifndef VCC_H
#define VCC_H

#include "Arduino.h"

class Vcc
{
  public:
    /**
     * Constructor
     *
     * @param correction  Correction factor, when reported Vcc is off from measured (externally) Vcc
     *                    (due to variations in bandgap voltage of +/- 0.1V)
     *                    Calculate as Vcc_measured/Vcc_reported. Defaults to 1.
     */
    Vcc( const float correction = 1.0 );

    /**
     * Retrieve current Vcc level.
     *
     * @return Current Vcc level, in Volts.
     */
    float Read_Volts(void);

    /**
     * Retrieve current Vcc level.
     *
     * @return Current Vcc level, in milivolts.
     */
    uint16_t Read_Volts_fast(void);

    /**
     * Retrieve current Vcc level. The total voltage range shall be passed
     * as low/high bound. For e.g. an alkaline AA battery this range can be set
     * to [0.6,..,1.5] Volts.
     *
     * @param range_min  Low bound to Vcc level range, in Volts.
```

```
61      * @param range_max  High bound to Vcc level range, in Volts.
62      * @param clip       When set, assures returned percentage is clipped to
    [0..100]% range.
63      * @return Current Vcc level, as percentage of expected Vcc level.
64      */
65     float Read_Perc(const float range_min = 0.0, const float range_max = 0.0, const
    boolean clip = true);
66
67   protected:
68     float m_correction;     /**< Correction factor, when reported Vcc is off. */
69 };
70
71 #endif
```